AFRL-RI-RS-TR-2016-009

TOWARDS THE UBIQUITOUS DEPLOYMENT OF DNSSEC

PARSONS GOVERNMENT SERVICES, INC

JANUARY 2016

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2016-009 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/ S /

FRANK H. BORN
Work Unit Manager

/ S /

WARREN H. DEBANY, JR
Technical Advisor
Information Exploitation and
Operations Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JAN 2016	FINAL TECHNICAL REPORT	JAN 2011 – JUN 2015

4. TITLE AND SUBTITLE

TOWARDS THE UBIQUITOUS DEPLOYMENT OF DNSSEC

5a. CONTRACT NUMBER
FA8750-11-C-0088

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

George R. Mundy

5d. PROJECT NUMBER
HS37

5e. TASK NUMBER
SP

5f. WORK UNIT NUMBER
AR

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Parsons Government Services, Inc
25531 Commercenter Dr STE 120
Lake Forest CA 92630-8874

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2016-009

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA# 88ABW-2016-0003
Date Cleared: 4 JAN 2016

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report describes the effort performed under contract number FA8750-11-C-0088 for supporting the ubiquitous deployment of DNSSEC. This effort included various sustainability, outreach and standardization activities; software development aimed at increasing the demand for DNSSEC and enabling DNSSEC validation support within end-applications; and, making the technology operationally robust and reliable.

15. SUBJECT TERMS
Domain Name Server, Security, DNSSEC

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**FRANK H. BORN**
U	U	U	UU	28	19b. TELEPHONE NUMBER (*Include area code*) **N/A**

Contents

SUMMARY **1**

1 INTRODUCTION **2**

2 METHODS, ASSUMPTIONS AND PROCEDURES **3**

2.1 STANDARDIZATION AND OUTREACH 3

2.1.1 Advancement of Standards *3*

2.1.2 Outreach *5*

2.2 SOFTWARE DEVELOPMENT 7

2.2.1 Enabling DNSSEC validating in applications *7*

2.2.2 Enabling DNSSEC on widely used platforms and Small Devices *9*

2.2.3 Extensions to DNS Provisioning Tools *11*

2.2.4 Improving DNSSEC Situational Awareness *13*

2.3 RESOURCE PROVISIONING 15

2.3.1 Certification and Accreditation *15*

2.3.2 Tool Dissemination *16*

3 RESULTS AND DISCUSSION **17**

4 LESSONS LEARNED **18**

5 CONCLUSIONS **20**

6 DIRECTIONS FOR FUTURE WORK **20**

7 REFERENCES **23**

8 LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS **24**

Summary

The Secure Extensions to the Domain Name System (DNSSEC) comprise of a set of protocol extensions that fix a critical weakness in the Domain Name System (DNS). However, since the technology pertains to a core component of the Internet infrastructure, deployment of DNSSEC on the Internet has met with a number of challenges. For instance, Zone operators must change their zone production workflow to ensure that the cryptographic assurances that DNSSEC provides are maintained, while applications must be made DNSSEC capable in order to ensure that the cryptographic attestations covering DNS data is actually validated. On account of the complexity involved, only the "expert" early adopters have typically adopted DNSSEC.

Our goal in this project was to help bridge the gap between the expert and the novice DNS operator through a combination of tools, outreach, standardization and education related activities. As part of our effort on this contract, we have made a number of contributions towards each area.

In the area of Standards advancement, we have co-authored Internet specifications that will help increase user demand through a particular application use case of DNSSEC. We have also co-authored specifications to increase the robustness of validators that must perform validation through non-standards compliant devices. We have also co-authored an RFC that helps automate certain DNSSEC operations that have traditionally been error prone, thus providing a solution to one of the deployment barriers associated with DNSSEC. Our outreach related activities included vendor discussions and presentations at various venues, including multiple Internet Corporation for Assigned Names and Numbers (ICANN) meetings.

We have built and made openly available a number of tools as part of the work performed on this contract. These include libraries that provide DNSSEC validation functionality; enabling validation within select applications, and particularly within a number of small devices; developing a number of other tools in order to make the task of DNS provisioning easier; as well as designing, implementing and distributing a framework for supporting troubleshooting and providing better situational awareness over DNSSEC and DNS operations.

Finally, this effort also included tasks related to management and operation of a Certified and Accredited environment to host the various resources that we made available to the community through the DNSSEC-Tools open source project and related resources.

1 Introduction

Making technological improvements to the Internet infrastructure have some unique challenges but these can be summed up through the following two observations. First, the technology is a core part of Internet fabric; thus, any enhancements must be backwards compatible so as to minimize the likelihood of any drastic and unwarranted effects on the existing infrastructure by the introduction of change. Second, Internet Infrastructure technologies are implemented by a variety of entities in a variety of ways and changes to the technology must account for evolutionary changes to a large and diverse number of environments. Any technological enhancements must account for devices that operate in unique environments and are administered by individuals who may have varying degree of knowledge about the technology.

These complexities associated with the adoption of technological innovation within the Internet core technologies are clearly evident in the deployment of DNSSEC. At the commencement of this contract, DNSSEC deployment was at a crossroads. A backwards-compatible specification for DNSSEC was finally available after a decade of steady progress and had even been implemented by some DNS vendors. A number of important zones, such as the Root and many Top Level Domains (TLDs), had also been signed and a small, but important, set of organizations had begun performing by-default DNSSEC validation. However, much of this progress was limited to early adopters. The cumulative level of deployment was still quite low and nowhere close to the proverbial "tipping point" that one normally associates with the diffusion of technology.

Our work in this contract was to help bridge the gap between early adopters and the subsequent adopters in order to help make DNSSEC ubiquitous. This involved standardization activities, outreach activities, the development of various software components and, various measurement and monitoring related activities. Finally, in compliance with the DHS policy, we configured, managed and operated a system that was compliant with the Federal Information Security Management Act of 2002 (commonly known as "FISMA") in order to make the various components of our solution widely available through the project website.

The funds available for this contract and the period of performance were truncated well short of the planned end date for the project. However, in this project's shortened lifetime, we have accomplished much and have learned even more. In this report we discuss our findings, observations and lessons learned from our effort at trying to make DNSSEC more ubiquitous.

2 Methods, Assumptions and Procedures

Our effort in this contract can be broadly divided into three categories. The first relates to standards development and outreach activities, the second relates to the technical components that we developed as part of the effort, while the third relates to the provisioning of services. We discuss these areas in greater detail below.

2.1 Standardization and Outreach

2.1.1 Advancement of Standards

Our standards related efforts were under three categories. One category was related to creating a validator Application Programming Interface (API) for use by DNSSEC-capable applications while another category related to automating some of the steps in the DNSSEC provisioning process. The third and, arguably, the most import category of our standards related activities had to do with the advancement of the DNS based Authentication of Named Entities (DANE) related specifications [DANE].

The DANE specification enables applications to use DNS as a means to provide additional assurance during the establishment of a secure TLS connection with a host. It helps in bootstrapping trust when the CA that issued the certificate is not well-known and it helps in selecting the basis for trust when the number of well-known CAs are many. DANE has often been referred to as the "killer app" for DNSSEC since it produces tangible benefits to the end user by reducing the costs associated with requesting certificates from CAs and through the potential of making ubiquitous encryption a reality.

In this contract we made a number of contributions toward the DANE standardization effort. During the initial stages of the standardization process we reviewed the DANE protocol specifications and contributed text to the draft. Once the specification work was complete, we progressed other drafts related to operational aspects of DANE. These included DANE TLSA operational guidance [DANEOP], which provides operational guidance to server operators to help ensure that clients will be able to authenticate a server's certificate chain via published TLSA records; and opportunistic TLS for SMTP via DANE [DANESMTP]. Both documents have been now submitted to IESG for Publication as Proposed Standards.

DANE has been continuing to draw interest in a number of other working groups. There are some very interesting possibilities in particular with respect to DANE and Session Initiation Protocol (SIP). However, it remains to be seen how quickly these enhancements are picked up and deployed.

The second component in our specifications-related effort was tied to automation of DNS service provisioning. DNSSEC adds certain complexities into the operator flow of operations. Through discussions with operators and through various DNSSEC-related workshops that we had conducted in the past, it became apparent that

automation was the key component that was likely to make DNSSEC more palatable to operators. As part of the work performed on this contract, we participated extensively in the proposal process for automating the management of secure delegations between parent and child zones through the CDS record. We also submitted a generic proposal to fix the long-standing operational problem with the parent's glue being out of sync with the child. Both documents were subsequently published as RFC specifications [RFC7344][RFC7477].

Finally, the third standards related area that we were actively involved in had to do with the API between an application and validation functionality provided by a validating stub resolver. We perceived this component as important, since a standardized API would make it easier for the application developer community to integrate DNSSEC within applications. During the project we engaged with various other name server vendors on various aspects of enabling DNSSEC validation on end-applications, including the need for an asynchronous API [VALAPI].

At the 83rd Internet Engineering Task Force (IETF) meeting we coordinated a meeting between various stakeholders in the application space in order to better understand the requirements for a DNSSEC validator API. A number of useful results emerged from these discussions but it became clear that the opinions concerning the needs of the API were quite fractured within the stakeholder community. On the one hand there was the need to support validation within existing applications and on the other was the desire to start from a clean state and build something more comprehensive. The getdns API [GETDNS] was an attempt to start with a clean slate by looking up information in the DNS using a modern asynchronous API. This proposal while useful in some respects also has some drawbacks such as its extensive use of particular data structures, which introduce additional overhead in languages that do not natively support these types.

A standardized API continues to be an important element for end-application validation support, especially now that more vendors are willing to adopt and include DANE support in their software products. However, it is not clear that the community has actually developed consensus around any particular specification. Since there are a number of DNSSEC parameters that an application could set, it would also be useful to have a standard expression on what these parameters ought to look like. However, currently, there is none.

We perceive robust fallback behavior from validators to be crucial to end-system adoption of DNSSEC validation, especially within non-compliant infrastructures. As part of work done on this project, we began documenting various types of DNSSEC deployment problems related to non-compliant infrastructure and potential detection and mitigation techniques for these problems [FALLBACK]. The document described ways that a validating host resolver could test a nearby resolver for use as a caching resource if it was DNSSEC-aware. It described tests that could be done, as well as mitigation techniques that could be used to circumvent certain problems. Our goal was to document how various test suites act and how multiple avoidance

techniques could be used in various libraries in order to document existing practice and help future implementations. This topic is important for the IETF DNS Operations Working Group and was subsequently adopted as a Working Group document. However, due to reduced resources that were available to us, we have been unable to progress this document within the IETF.

Other areas that we actively contributed towards within the standardization space include dns-privacy, where the emphasis was on link encryption and query minimization as a means to address concerns surrounding pervasive monitoring. In addition, Opportunistic Security (OS) [RFC7435], is yet another specification where the use of DNSSEC as a privacy enhancing mechanism is being discussed. The OS idea originated from the DANE for SMTP draft, which we co-authored as part of our specifications related effort.

The standards related process for DNSSEC is expected to continue until such time that DNSSEC attains widespread deployment and use on the Internet to ensure that DNSSEC continues to work in harmony with other standardization activities in the IETF. As an example, the "therightkey" Working Group developed the 'Certificate Transparency' approach as a detection-based control for compromised CAs. While the threat models are very different for the certificate world and DNSSEC, at the time of the development of the specification not all aspects of the CT technology were consistent with DNSSEC. As part of our effort on this contract, we engaged in various discussions with the relevant set of participants so that DNSSEC and DANE were given due consideration in the design stage. It is essential that such discussions continue in order to ensure that future applications that affect the Domain Name System do not work at cross-purpose with DNSSEC.

2.1.2 Outreach

While standards advancement were necessary to enable DNSSEC vendors to incorporate the new features into their products, outreach was necessary to ensure that vendors were committed to adopting the standards, and to ensure that operators actually made use of the new features. This section describes some of the outreach activities that we carried out in this area of the effort.

2.1.2.1 Program Coordination

We met with large vendors such as Mozilla and Microsoft early in the project and encouraged them to deploy DNSSEC in their products. Microsoft has made significant headway in the adoption of DNSSEC in its server products, which has made DNSSEC deployment within environments that are largely Windows-centric a real possibility. However, there are still some issues that persist within corporate environments that have not been resolved completely. Foremost in this list is the handling of validation within split-view DNS environments.

As part of some of our earlier work, our team had developed a draft specification for split view DNSSEC [SPLITVIEW]. As part of the current contract we began to look at this problem from a monitoring perspective in order to provide the administrator with an early warning notification when there was a name collision and when it was likely to result in DNSSEC validation inconsistencies. While some of that monitoring framework is available in our tool suite, we did not have sufficient resources to extend the tool for supporting the split view use case. However, it is likely that with greater DNSSEC uptake within corporate environments, the need for best practices and tools that make split-views easier to manage will become increasingly important.

In order to encourage adoption by diverse operator groups, we engaged with industry at multiple venues. We participated in the DNSSEC workshops at FOSE and at the Blackhat conference where we presented some of our work and represented the DNSSEC Deployment Initiative at the vendor booth. We created data sheets for the DNSSEC-Deployment effort and the DNSSEC-Tools project, outlining various technical resources that the team has provided in support of DNSSEC deployment. We also made our LiveCD for DNSSEC-Tools available to various operators in order to enable them to evaluate our open-source tools easily and to encourage them to use our tools for various training events. Finally, we also attended various NANOG meetings where we interacted with a number of DNS Operators and obtained their feedback on the types of tools they would need to implement DNSSEC in their zones.

2.1.2.2 Publications

Our outreach activities also included various information dissemination activities.

We shared, through formal publications, various insights that we had gained through the process of building DNSSEC-capable applications and tools to the operator and developer communities. The paper “Enabling DNSSEC in Open-Source Applications” [SATIN2011] described our efforts at enabling DNSSEC within a number of applications including the Mozilla Firefox browser, the Postfix message transfer agent (MTA) and OpenSSH. Another paper, “DNSSEC Automation and Monitoring” [DNSEASY] outlined a number of considerations related to automating DNSSEC operations and some of the tools available in the DNSSEC-Tools suite that support various automation and monitoring needs.

We also used some of the new media approaches for spreading our message, such as through blog articles and through white papers.

2.1.2.3 Global Operator Engagement

DNSSEC is a global technology. A local solution to global problem would fail to leverage any network level externalities that could be generated by deployment forces outside a particular region. Thus, in keeping with this principle, we engaged with operators throughout the world, trying to not only raise awareness about the need for DNSSEC but also trying to recognize any operational idiosyncrasies that could become deployment showstoppers.

The primary forum for global operator engagement was the DNSSEC plenary session at various ICANN meetings. As part of the work performed on this contract, we led the DNSSEC tutorial and the DNSSEC full day session at ICANN meetings where we discussed various topics related to ongoing deployment in the registry, registrar and application space and to get regional perspectives on how deployment could be furthered. The regional perspectives also helped us gain an understanding on what governmental and industry-level policies helped adoption and where the gaps existed. For example, at the ICANN 44 meeting in Prague in addition to the main DNSSEC session at the ICANN meeting, we also held an implementers session in order to facilitate the cross sharing of deployment experiences and best practices between various DNSSEC adopters. We held a discussion with a DNSSEC adopter on the current areas that, according to the adopter, lacked in sufficient deployment guidance. As a result of that discussion, we identified the need and developed an initial requirements analysis for a "DNSSEC Deployment Wizard" – an online tool that could be used to build custom deployment plans for different types of adopters.

2.2 Software Development

Our tool development effort in this contract was focused in four different areas that offered significant promise of hastening DNSSEC adoption. The first was enabling DNSSEC within select applications; the second was extending validation capabilities to small devices and other widely used platforms; the third was geared towards making provisioning of DNSSEC easier so as to seamlessly support various use-cases that emerged as DNSSEC got more widely adopted by the registrar and DNS hosting service communities; while the fourth area relates to helping the operator have improved situational awareness over the DNS resources that they operate. We discuss our effort in each of these areas below.

2.2.1 Enabling DNSSEC validating in applications

The libval library in DNSSEC-Tools is the core library that enables validation in a number of end-applications. One of the limitations of this library was its inability to asynchronously resolve and validate responses to various queries, this feature was required in order to instrument certain applications to be DNSSEC capable. In order to resolve this issue, we developed an asynchronous API and modified our validator library so that it could send DNS requests, validate answers and return results back to the application in an asynchronous manner. We found a significant speed-up of

query resolution using this asynchronous capability. This was important since this speed-up has the potential of offsetting some of the additional burdens DNSSEC imposes. The addition of this asynchronous capability had thus become one of the main enablers of DNSSEC validation on query intensive end-applications such as web browsers.

In order to make DNSSEC validation more robust we also implemented fallback logic in our validator library so that the library would attempt to multiple sizes using EDNS0 (the extension mechanism for DNS) and recursively fetch an answer from the root name servers if the answer returned from a local caching name server was bad or could not be validated. This feature gave applications the opportunity to recover from failure scenarios when the recursive name server was lying or was non-conformant with the DNSSEC specifications while also allowing it to use a resolver cache where possible.

We also made changes that enabled our validator library to be used within self-contained sensor environments. We added support for hard-coded validator configuration information that could be used if other configuration data was not found and added initial Transaction Signature (TSIG) support in libval so that queries could be sent to name servers that only supported authenticated lookups. We added a flag to ignore data in the answer cache while trying to resolve a name and also added support for the retry and timeout policy knobs in order to allow query retries to be more deterministic. We also modified the zonecut processing logic so that the validator looked for zonecuts based on NS records rather than Start Of Authority records (SOAs) and added a new feature to conditionally check for all signatures within a given assertion structure even when one that validates had already been found. This feature proved to be very useful in anticipating and debugging validation failures.

As described earlier, DANE represents a key use case for DNSSEC and is likely to propel the use of DNSSEC across a wide spectrum of applications. In light of this, we added initial support for DANE in our validator library and developed wrapper tools to check an X509 cert provided over the SSL connection against the TLSA record. There were a number of implementation related corner cases that had to be taken into account and some ambiguity; particularly concerning how openssl (the underlying crypto library) performed its X509 checks. It is clear that while DANE represents an important positive development for DNSSEC, its implementation still suffers from a lack of diversity. There is only a very small pool of implementations that currently exist for DANE today and more inconsistencies in implementation details are expected to occur over time.

Extensions and modifications we made to our validator library add and expand the DNSSEC validation capability of a number of widely used applications such as NetSNMP and Curl. In response to a request from the CeroWRT project developers, we enabled DNSSEC-support in the Network Time Protocol (NTP) code-base and submitted this patch to the NTP developers. The problem the CeroWRT project was

trying to solve using our patch was the circular dependency on time for certain devices and the dependency on DNS. With a badly offset time, DNSSEC validation would fail and without the means to validate responses, the system using the CeroWRT software would not be able to contact NTP servers to make adjustments to their local time. Our software provided the ability for NTP, when supplied with the appropriate option, to ignore signature inception and expiration times till such time that the system time was within some threshold value.

As part of our effort under this contract we also worked towards enabling local DNSSEC validation within Firefox using the asynchronous API. There were a number of aspects that made this problem non-trivial, including complexity of the Firefox code, the difficulty in keeping our code changes in sync with the most recent version of Firefox, maintaining the build pre-requisites, and making the changes available in a manner that could be tested out by the community.

In order to address the issue of binary distribution, we developed and packaged Bloodhound, a modified and re-branded version of the Mozilla Firefox browser. Bloodhound provides local DNSSEC validation of all DNS names on a page and the DANE protocol. The difficulty in keeping the code and build environment up to date was evident from our ability to only support a limited of platforms with Bloodhound. We made an initial version of a Bloodhound Linux binary available, and multiple versions of the OSX version of Bloodhound available but we did not have sufficient resources to complete our Windows port of Bloodhound. While other DNSSEC browser capabilities are available to validate the URL address bar name, Bloodhound remains the only web browser capable of validating all DNS lookups on web pages, these DNS lookups can sometimes be in the hundreds on a single web page and are all subject to hijack.

There were a number of changes that we had to make to the Firefox code base, including making significant changes to Mozilla's resolver library so that it could support asynchronous lookups and to support the use of the DANE protocol to bootstrap the validation of self-signed certificates for DANE-enabled websites. The crypto library used to implement transport-level security within Firefox lacked sufficient documentation; so, without the active help of Mozilla developers, we were only able to mark our code as experimental. However, we continued to encourage the browser community to enable DNSSEC and DANE support within their products and made all our application patches freely available via our website.

2.2.2 Enabling DNSSEC on widely used platforms and Small Devices

We realized at the outset that merely retrofitting legacy applications to be DNSSEC-capable was not going to be sufficient to spur DNSSEC adoption within end-systems. Thus, we explored facilitating deployment at the leading edge of technology through DNSSEC deployment on portable devices and other widely used platforms.

We ported our validator library to multiple platforms including Windows, IOS, and Android based platforms as well as writing an initial proof-of-concept application showing validation functionality on an iPhone "simulator" device. We began looking at the possibility for including local validation support on various mobile hand-held platforms and built many sample QT-based DNSSEC validating applications that were capable of being run on a variety of platforms. (QT is a cross-platform framework that enables the creation of applications that are compatible for a variety of platforms including certain phones)

We were also able to enable DNSSEC in the base stub resolver of the QT framework. In addition, certain widely-used applications such as the Opera browser are also based on QT thus enabling DNSSEC in the QT layer provided the possibility of making these applications DNSSEC capable. We also worked with the QT developers towards building a generic query class with support for DNSSEC validation. This was to allow easy querying for types such as TLSA.

A number of our QT based tools pertained to enabling the operator to perform better troubleshooting of DNSSEC error conditions and inconsistencies. We used the QT framework to build a number of DNSSEC applications for the end-system, including an application to perform local validation of a domain name and display its result; an application to test the local resolver environment for DNSSEC capability and whether it could be routed around if the resolver is broken; a system-tray application that notified the user whenever the validator encountered an error; and an application to provide a visual depiction of the validator log messages and authentication chains.

The last tool, which we called DNSSEC-Nodes proved to be very useful in troubleshooting and as an education aide. In order to bring the value provided by this tool to other users we extended our DNSSEC-Nodes application in a number of ways:

- We added support to parse BIND log files and display the authentication chain graph based on those results.
- We added the capability to sniff packets on the wire and to read TCPdump files and build validation graphs from the observed packets.
- We added the capability to initiate validation for a given name from within the DNSSEC-Nodes utility and to build and display the validation tree for that name.
- We also enabled a user to click on a box to highlight a given node and the arrows around it. This feature, while useful for troubleshooting, also served as a useful DNSSEC instruction aid.
- We built an extensive filter editor enable the user to tailor the look of a graph to color-code, set the levels of nodes based on their names, status and data types, and added the ability to center the graph on any given node.

- We added support for many data types in the data view and added support for many command-line options.

The combination of these pieces of functionality enabled the administrators to gain a better idea about the types of DNSSEC errors that were occurring and the nature of the problem.

DNSSEC-Check was another application that we extended using the QT framework. We ported the tool to run on a number of platforms including certain mobile platforms such as Android, Maemo, Meego, and Harmattan. We also made the tool available in the Android marketplace. The tool was also updated to submit hashed IP addresses to a central DNSSEC-Tools repository and aggregation summaries of the data collected are being generated. We demonstrated the value of this tool in various operator forums and encouraged users to try the tool and help measure the deployment of DNSSEC.

In addition to basic measurement of DNSSEC, we updated the DNSSEC-Check tool to include a letter grade assigned to each resolver based on the results of the tests performed, where the letter grade was in direct correspondence to the FCC Communications Security, Reliability and Interoperability Council's (CSRIC) recommendations on DNSSEC for Internet Service Providers (ISPs) [CSRIC]. The DNSSEC-Check tool thus provided a relatively light-weight approach for verifying if ISPs were in compliance with the CSIRIC recommendations, especially in cases where the user was mobile.

2.2.3 Extensions to DNS Provisioning Tools

DNSSEC represents a shift to the normal workflow for a DNS administrator. In cases where the operator neither has the expertise nor the time to integrate these tools into their operating environments, DNSSEC is yet another sub-system that operators will be forced to deal with separately. DNSSEC is less forgiving than legacy DNS and demands certain good administration practices. DNSSEC operator errors can easily result in service unavailability in cases where vanilla DNS would still work.
Tools for simplifying DNSSEC operations are clearly essential for ensuring that operator errors are minimized to the largest extent possible and to reduce the temptation for zone operators to completely disable DNSSEC services for their user population. In order to identify the relevant set of improvements to make to our DNS Provisioning tools, we relied on best practices and operator feedback.

We completed a review of the RFC4641bis “DNSSEC Operational Practices” [RFC6781] document where we considered how each operation described in this document could impact large-scale deployments of DNSSEC. In parallel, we developed a list of events to monitor, based on the various timing dependencies, associated with different DNSSEC operations. We then surveyed a number of

candidate operator tools to instrument in order to simplify and make DNSSEC operations for the provisioning end more widely available.

As part of addressing some of the gaps that we identified, we completed an initial implementation for DNSSEC support in Webmin and provided the patch to the Webmin developers, who integrated it into their mainline distribution. The DNSSEC-Tools patch for webmin enables the zone administrator to use the tools from the DNSSEC-Tools suite to manage DNSSEC operations on their zones using the Webmin front-end. One of the features added was allowing the zone maintainer to obtain a consolidated view of the different rollover phases for each zone being managed. This allowed the zone administrator to quickly determine which zones needed administrator intervention in order to complete their key rollover operation properly.

During some of the operator related conferences, we had multiple useful discussions with various name server operators and used their input to build various improvements to our tool set. Some of these features were specifically geared towards making DNSSEC adoption easier for entities that provided DNS hosting services since deployment at such locations is likely to propagate to many customers simultaneously.

In zonesigner, for example, we added the threshold option to specify a signing threshold. This simplified the automation of re-signing a large number of zones when the administrator wished to script the process within a scheduling daemon such as cron. We enhanced our zonesigner utility to support "out-of-band" signing so that operators could force a safe re-sign of their zone even when the zone was in one of the wait intervals within a key rollover operation. We completed our support for zone groups, which allows a collection of zones to be controlled as a group, rather than controlling each of those zones individually. Additionally, we added a "threshold" option to enable the operator to manage multiple zones with different signature inception and expiration times. Zonesigner can now be invoked as often as needed but will only re-sign the zones where the signatures are nearing expiration.

We also made operator driven changes to rollerd where we added support for multiple instances of rollerd and added a new "dtrealms" feature that enabled an operator to manage a group of distinct rollover environments running on a single host. We also added a new "autosign" feature that would cause rollerd to examine if a zonefile has been modified more recently than its signed version and, if so, re-sign the zone file. We also added phase-specific commands and additional commands (via rollctl) to allow greater control over zone rollover actions. Finally, we improved the manner in which rollerd identifies the zones being managed and are consequently able to support thousands of zones with a single rollerd instance when previously only a few hundred zones could be managed with a single instance.

There are a substantial number of other capabilities needed for the provisioning activities required for successful deployment of DNSSEC but limited resources prevented their implementation. For example, the recently concluded IETF activity defining specifications for automated mechanisms to keep parent and child delegation information consistent has not been implemented in DNSSEC-Tools or any other openly available reference implementation. Use of the specified approach will significantly reduce the number of operational errors that have occurred in practice and lack of these tools is discouraging the deployment of DNSSEC beyond top level domains.

2.2.4 Improving DNSSEC Situational Awareness

While the use of automated tools was expected to reduce the number of operational errors associated with DNSSEC deployment, we also realized that good monitoring tools were going to be essential in ensuring that problems were detected early and fixed ahead of widespread outages. Through the work that we performed on this contract, we made significant headway into making available an open source implementation of a DNSSEC monitoring framework.

We began by looking at ways to add DNSSEC monitoring capability into existing monitoring frameworks. Nagios is a widely implemented tool that is used for IT infrastructure monitoring. We developed modules for Nagios to support DNSSEC related event monitoring and developed a new module that tied the zone checking functionality of our "donuts" tool with the Nagios engine to provide a zone operator with periodic notifications of zone errors. Since donuts rules are highly customizable, this feature gave the zone operator the ability to tune the types of errors they wanted to be notified of to a fine level of granularity.

We also added new sensor capabilities to integrate the functionality provided by the trustman tool into the Nagios framework. This allows a zone administrator or a name server administrator (such as an ISP) to observe the trust anchor state from trustman's vantage point and be notified of inconsistencies that could result in validation errors. Finally, we created a Nagios module that allowed us to monitor the status of key rollovers performed by our rollerd utility and developed a simple tool capable of testing signature expirations on existing zones.

Zabbix is another framework that is widely used for monitoring IT infrastructure services. In order to extend some of the DNSSEC monitoring capabilities to the Zabbix environment, we developed a simple configuration that could be used to monitor DNSSEC-Tools rollover using the Zabbix framework.

Since DNS is essentially a distributed service, it is useful to distinguish between the health of the set of name servers that host the DNS zones and the health of the name service as viewed by users from across the Internet. The user centric view is

affected by a number of variables such as middle boxes and recursive name servers, which are typically not included in the name server-centric view. In addition to the monitoring enhancements in Nagios and Zabbix, we worked towards the creation of a DNSSEC-enabled operator front-end so that operators would have a unified administrative handle and, consequently, improved situational awareness over their DNS operations. We named this tool 'Owl'.

The basic idea of the Owl monitoring system is that of monitoring the DNS from multiple sensor locations and aggregating the results in a manner that would allow the DNS administrator to assess the health of their DNS service from multiple vantage points. The Owl Monitoring System uses timed DNS queries to monitor basic network functionality. The system consists of a manager host and a set of sensor hosts. The Owl sensors perform periodic DNS queries and report to the Owl manager the time taken for each query. Over time, this shows the responsiveness of the DNS infrastructure.

We incorporated support for DNSSEC-specific monitoring within the sensor and manager code and held extensive brainstorming sessions to discuss ways in which we could present operators with a useful array of indicators that they might want to monitor over time. We also enhanced some of the supporting tools used within Owl so that operators could query for data from the desired name servers and perform the correct set of checks from different sensor locations.

In order to understand the resource requirements for Owl, we added a plugin to check disk usage as well as other checks of system usage by the Owl sensor and management station. The plugin provides important metrics on how well the system scales with thousands of sensors and helps define operational requirements for large Owl installations. As a result of some of the data we collected on the manager's processing overhead, we also added the ability for owl sensors to segment the data that it was sending back to the manager. We also continued to improve donuts, our DNS checking utility, so that it could run its tests on piecemeal data returned by various sensor instances, providing the building blocks necessary to enable better monitoring of DNS and DNSSEC errors as they occurred at various points on the Internet.

The development of the owl framework required some changes to some of our other tools. As an example, we had to add support for dynamic policies in our validator library so that applications could create validator contexts with a custom resolver and validator policy in order to send specific types of queries to specific name servers based on the sensor configuration file. We also added asynchronous lookup support to the validator Perl module so that sensor modules could send out streams of queries without any delay between lookup operations. Also, we made a number of infrastructure changes in order to facilitate the easy creation of new sensors and to ensure that the availability of data collected from sensors was reasonably protected. We explored possibilities of setting up additional Owl sensors within a corporate

environment in order to get a better understanding of how the system could provide benefit to administrators within a corporate network setting.

The number of uses of the Owl monitoring system are many. While we primarily envision it being used in the context of zone operators checking the health of their name service, we were also able to use the framework to monitor response times for the Root name servers during the period of the rumored server takedown by Anonymous, and used the framework to analyze the K-root server's response time following reports of increased query loads at the K-root.

Currently, DNS is monitored by administrators through a potpourri of scripts. Owl has the potential of making the set of monitoring operations more standard, thus making DNSSEC operations much more robust. However, there are still a number of sensor modules that need to be written in order for Owl to reach its true potential as a production-level DNSSEC monitoring tool. This is something that would be certainly worth pursuing as part of future work.

2.3 Resource Provisioning

2.3.1 Certification and Accreditation

Another portion of the effort under this contract was the management and operation of the infrastructure necessary to host various DNSSEC deployment resources within a Certified and Accredited (C&A) environment.

The different tasks that we performed as part of C&A activities include the following:

- We performed required updates to keep the system protected against current web-based threats, performed various continuous monitoring tasks, and updated the system documentation accordingly.
- We worked with other initiative partners to clean up content on the dnssec-deployment.org website, updated the system configuration to enable certain types of posts, and added a new calendar plugin to the blog.
- We updated the System Security Plan based on the HQ review of the C&A documents for the RDI system.
- We went through a security test and evaluation of the system with the DHS S&T compliance team and resolved all Plan of Action and Milestones (POA&Ms) that were identified for the RDI system.
- We requested, and were grated, a waiver for using SSL certificates issued by a commercial CA.
- We worked with the compliance office on the need for a Privacy Policy Notice and a Comment Policy on the websites on the RDI system.
- We performed annual Contingency Plan table-top tests for the RDI system and documented our results.

- We responded to various data calls from the compliance office. One was related to our system inventory; another was a domain survey for the websites hosted on the system; an additional one was a DHS Chief Information Security Officer (CISO) data call pertaining to identifying Component Privileged Users for the RDI system
- We attended the S&T Annual Cyber Security Stand-Down (STACSS) as part of the continuous training related to performing the Information System Security Officer (ISSO) function for the RDI system and reviewed material associated with the required privileged user training for the RDI system.

All Certification and Activities for the RDI system concluded in January, 2014. All resources tied to the DNSSEC-Tools project was transferred to Parsons Corporation, and has since been managed as an independent open-source project. All resources that were part of the DNSSEC-Deployment effort, including the various mailing lists, were transitioned; first to Parsons Corp, and then to the Internet Society (ISOC) in order to augment ISOC's DNSSEC-outreach efforts.

Parsons also signed a memorandum of understanding (MOU) with the Internet Society to collaborate on initiatives to promote the global deployment of DNSSEC. From the announcement, "the MOU is a formal endorsement of the cooperative arrangement between the Internet Society's Deploy360 Programme and the DNSSEC Deployment Initiative".

2.3.2 Tool Dissemination

We used multiple approaches to make various tools available to a number of operators. We released multiple versions of the DNSSEC-Tools suite and announced them both on the project mailing list as well as on social media. We packaged a number of our tools for the provisioning activities and the validating activities into a Fedora "spin", allowing us to create a DNSSEC-Tools Live CD. The LiveCD included the various QT tools and a DNSSEC-Tools demo framework that allowed operators to examine (and install to hard-disk if desired) the different DNSSEC-Tools components.

We also created instructional videos that illustrate how an operator can get started with using some of our tools for managing their zones. The instructional videos are available on YouTube. We also created a poster highlighting some of the tools in the DNSSEC-Tools suite.

Finally, we created a new rpm repository 'dnssec-extras' to facilitate the distribution of various DNSSEC-capable application binaries, including Bloodhound, on Linux-based platforms.

3 Results and Discussion

This effort resulted in a number of useful contributions towards advancing the state of DNSSEC Deployment on the Internet. We made our tools freely available in order to simplify the provisioning of DNSSEC and make deployment ubiquitous. Similarly, our licensing terms encourage third-party use and we have seen successful technology transition to core DNS products with significant potential for future commercial offerings. We summarize some of our main contributions below:

Our work has facilitated the growth of a vibrant worldwide community supporting DNSSEC deployment. We performed a number of outreach activities including conducting training sessions and leading the DNSSEC workshop sessions at many ICANN meetings. We have thus far conducted DNSSEC workshops at each of the three major ICANN meetings since 2005 (ICANN 22). On account of the various outreach activities that we have carried out in collaboration with other deployment partners around the world, there is now a significant and growing number of TLDs that have been signed, and a number of operators and organizations that widely recognize the need to deploy and use DNSSEC. Additionally, the ICANN new generic Top Level Domain (gTLD) program now requires new gTLD's to be DNSSEC signed when they are initially deployed. There is also now a small, but important, set of organizations performing by-default DNSSEC validation.

Through our specifications-related work, we have provided a much needed, essential basis for secure bootstrapping of new applications using standardized, open DNSSEC technology. We co-authored two IETF specifications related to DANE, one relating to operational guidance and implementation suggestions for DANE, and the other pertaining to opportunistic TLS for SMTP. We also added initial DANE capability to our validating library in order to make the technology easy to use and accessible. We have also extended the DANE capability to a number of applications including Firefox and curl. In order to clearly distinguish our contributions from the official Firefox distribution we packaged and released "Bloodhound", a re-branded version of the Firefox browser that performs DNSSEC validation for all DNS queries (rather than just the URL address bar) and contains initial support for the DANE protocol.

We also worked on building support for DNSSEC and DANE within the QT framework. The QT framework allows developers to build portable applications for a variety of platforms, including a number of small devices. Many widely used applications, such as Google Earth, the Blackberry 10 operating system, and the entire set of K Desktop Environment (KDE) windowing system applications are based on QT. Enabling DNSSEC and DANE in the QT layer has the potential to make these applications DNSSEC and DANE capable as well.

We worked with multiple organizations to improve the robustness of DNS and DNSSEC operations and security, and incorporated many of our findings into our

open source tool offerings. We authored two IETF specifications in this space, one relating to synchronization between parent and child entities, and the other relating to detection and mitigation techniques for DNSSEC aware resolvers that were operating within a non-compliant infrastructure.

We built a DNS and DNSSEC monitoring system called "Owl", which is a distributed sensor system designed to detect the health of a deployed DNS system and a set of DNSSEC-specific monitors for communicating detected zone state and error information to the operator. We also built a prototype instance of the Owl sensor on a Raspberry Pi device running Debian Linux to verify the viability of easily standing up sensor nodes on devices that with a small form-factor and relatively modest computational power.

We made a number of enhancements to our DNSSEC troubleshooting and visualization tools in order to support validation tree graphing, on-the-wire traffic display, pcap packet-capture display as well as increased data logging. Some of the visualization capabilities can be further developed as commercial offerings that could provide an enterprise with better situational awareness over their DNS operations.

We also configured and managed a Certified and Accredited environment where we hosted the open source software that we created, and operated the various mailing lists associated with the deployment effort. We also created a new rpm repository, "dnssec-extras", to facilitate the distribution of various DNSSEC-capable application binaries, including Bloodhound, on Linux-based platforms.

4 Lessons Learned

Retrofitting changes to core Internet technologies that have known deficiencies is extremely important, but it takes a significant amount of time. There are a number of reasons this retrofitting requires time, including the ubiquity of the technology being replaced; the hidden assumptions associated with the use of the technology in particular environments; and the constant evolution of technology itself. DNSSEC has many additional complexities in the form of the Registrant-Registrar-Registry ecosystem, the various DNS hosting providers and the different devices that are capable of participating in the name lookup process. During the course of this project we have continued to gain deeper insight into the multiple moving pieces associated with DNSSEC and we summarize some of the key ones below.

Deployment of technology encompasses not just the technology, but also the various interests that are deeply intertwined within the Internet infrastructure machinery. We observed this in our attempt to integrate DNSSEC within Internet browsers, and also when trying to develop specifications for automating parent to child communication of secure delegation information. In both cases, there are business

relationships that play a large part in deciding how the technology is likely to be adopted by the different entities concerned.

Tools are an essential component in making any new Infrastructure technology more palatable to the operator. However there are a variety of tools required and the same tool often needs to be modified multiple times in order to account for new deployment use cases, new operator needs and sometimes evolving best practices. Operators need tools that are robust and robustness of tools can be impacted in many ways, sometimes outside the control of the tool developer themselves.

For example, some of our zone maintenance tools relied on existing third-party software modules. When the tool developers of this third-party software changed their API without any backwards compatibility, it affected certain operators who were no longer able to use the tool that we made available. It is important that support for open source software that facilitates DNSSEC deployment continue well into deployment of the technology, in order to provide operators with some degree of assurance that the software will continue to operate reliably even in the face of breakage caused by such external dependencies.

It is also important for DNSSEC validation to be robust. There is currently a business cost for turning on DNSSEC and the cost of false positives is higher than what browser vendors are willing to handle. Devising good fallback mechanisms might provide the means to make DNSSEC more palatable to the browser community. Technologies such as DANE generate user demand and are therefore strong drivers for DNSSEC deployment. However, the technology also competes with the interests of certain CA vendors. We observed that CA operators in general were less opposed to DANE when it was used to augmented the X509 framework rather than establishing a new trust infrastructure altogether. Further, unless there's a clear approach on how the reliability of DNSSEC can be improved under certain conditions and how the registrant provisioning process could be made more secure it appears that the CA and browser community would prefer alternatives such the Certificate Transparency approach for detecting miss-issued certificates rather than DNSSEC and DANE based mechanisms.

We also recognized that, while DNSSEC as a technology, is and should be transparent to the end user, there are a number of errors and failures on the part of DNS operator errors are likely to impact the end user in a myriad of ways. Ensuring that users are made aware of the nature of the problem, identifying the entity that might be able to resolve the problem, and most importantly, helping the user differentiate between transient errors and legitimate DNS spoofing attacks is likely going to be very important as deployment grows. The importance of using non-traditional notification methods is likely to grow, especially in the manner that some ISPs leverage social-media for providing their user base with timely information about ongoing DNSSEC-related issues.

5 Conclusions

Our goal in this project was to develop a number of components that would take us towards a state of ubiquitous DNSSEC Deployment.

We used a four-pronged approach in the effort to make DNSSEC Deployment ubiquitous. The first was aimed towards increasing the demand for DNSSEC; the second was to build some of the technology components that enabled DNSSEC operations and validation; the third was the sustainability, outreach and standardization component; and the fourth was making the technology operationally robust and reliable.

We have made a number of contributions to DNSSEC deployment as we have summarized throughout this report. As a consequence of some of the specifications related work that we supported there have been a number of promising developments in the DNSSEC application space, particularly in relation to DANE, and these are likely to spur further adoption within some application developer communities. In addition, we have built a suite of tools for the DNS operator, and a new DNS monitoring framework that has immense potential to make DNSSEC operations more robust.

While we have made a number of strides towards making DNSSEC more usable, the technology is still far from being ubiquitous. There are still a number of deployment related barriers associated with DNSSEC and DNSSEC Deployment at the provisioning end is only occurring at a modest pace. With no overwhelming adoption at the provisioning end, DNSSEC deployment is likely to stagnate unless demand is increased at the user end. Increasing the robustness of operations, especially for operators who are not DNSSEC savvy, is also important.

We began our effort on this project by observing that DNSSEC Deployment was at a crossroads. At the end of the project we are at yet another such crossroad. The challenges associated with deploying technologies that relate to the Internet core are different from those associated with other technologies. Without the existence of strong demand, a lot more active advocacy of the technology is required in order to ensure that deployment is nurtured and sustained. Currently there is some coordinated effort; it remains to be seen whether this will provide the necessary impetus for continuing ongoing DNSSEC deployment.

6 Directions for Future Work

While we have taken a number of strides in advancing the state of DNSSEC Deployment through the work performed on this contract, some gaps still remain. We see a number of possibilities in which the work that we performed on this contract can be enhanced and extended.

As discussed above, our work has resulted in several new tools that assist users during troubleshooting and monitoring operations. However, a lot more is required in order to improve the robustness of such monitoring utilities. We still see numerous instances of zone operators breaking their DNS availability on account of operational errors so there is also a large education component that is needed to ensure that operators make use of the various resources available to them, and do so in an effective manner.

In cases where DNS breakage results it must be easy for an operator to recover operations. Given the numerous time dependencies that affect DNSSEC operations, some knowledge of the DNSSEC and DNS protocol is necessary. However it is clear that for many operators DNSSEC is still much of an unknown quantity. So long as the perception of DNSSEC being a complex technology persists, there will be strong motivation for operators to completely disable DNSSEC as soon as they encounter any breakage condition. While the use of "negative trust anchors" will help some operators, in order to sustain deployment, tools that hand-hold operators through the recovery process are also essential. Such tools do not exist currently.

On the DNSSEC provisioning side, no registrar open source implementation exists to support DNSSEC on both user and registry sides. Significant conflict still exists surrounding best practice for submitting DNSKEY or DS records to a parent, and a wide set of EPP extensions are not well supported.

The occurrences of DNS based amplification attacks have also seen a rise in the recent months. The common variant of the attack involves a spoofed ANY query sent to an authoritative name server to request a larger response to the victim IP address. While this is not a DNSSEC related problem, it is important that standard practices emerge to thwart this type of attack without disabling DNSSEC.

On the validation side, only a few country ISPs do almost all of DNSSEC validation on the Internet. Validation by enterprises is essentially non-existent. For deployment on the validation side to grow, the demand for DNSSEC needs to increase. DNSSEC validation by users or enterprises can be driven by DANE and other emerging capabilities. However there are only few applications that implement DANE. In particular, Bloodhound is currently the only browser that supports both DNSSEC for all queries and DANE. While we have made our application patches available to Mozilla, most browser vendors do not appear to have any plans to support DNSSEC. DNSSEC validation on end-hosts is similarly sparse with only a handful of distributions shipping with a local instance of a validating resolver.

There are also other barriers that hamper DNSSEC adoption on the validation side. Many local environments are hostile towards DNSSEC in that middle-boxes may prevent DNSSEC checks from being completed. Delays and validation failures are likely to discourage use of DNSSEC and various inconsistent fallback approaches for validation failures are likely to discourage DNSSEC use and lower demand. Through

our work, we have started to outline a standard approach for fallback and recovery within validators but that work is not yet complete.

Earlier experience has shown that there will be new barriers to adoption that will emerge as DNSSEC is adopted within small devices. The impact of enabling DNSSEC validation on millions of smart-phones and tablets as well as in embedded devices has not been analyzed in depth but will certainly require further investigation. Further, automated key and algorithm rollover will be essential for end system deployment. Since the root key has not yet been changed and there is no protocol or procedure identified for changing the required DNSSEC algorithm, these activities will require significant research and tools support.

7 References

- [DANE] P. Hoffman et al. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, Aug 2012
- [VALAPI] S. Krishnaswamy et al. DNSSEC Validator API. IETF Work in Progress, draft-hayatnagarkar-dnsext-validator-api-09. Mar 2012
- [GETDNS] P. Hoffman, Description of the getdns API. https://getdnsapi.net/spec.html accessed Jun 2015
- [FALLBACK] Hardaker et al, DNSSEC Roadblock Avoidance, IETF Work in Progress, draft-ietf-dnsop-dnssec-roadblock-avoidance-01. Sep 2014.
- [DANEOP] Dukhovni et al, Updates to and Operational Guidance for the DANE Protocol, IETF Work in Progress, draft-ietf-dane-ops. June 2015
- [DANESMTP] Dukhovni et al, SMTP security via opportunistic DANE TLS, IETF Work in Progress, draft-ietf-dane-smtp-with-dane. April 2015
- [RFC7344] W. Kumari et al, Automating DNSSEC Delegation Trust Maintenance, RFC 7344. Sept 2014
- [RFC7477] W. Hardaker, Child-to-Parent Synchronization in DNS, RFC 7477, March 2015.
- [RFC7435] V. Dukhovni, Opportunistic Security: Some Protection Most of the Time, RFC 7435, Dec 2014.
- [SPLITVIEW] S. Krishnaswamy, DNSSEC Split Views. IETF Work in Progress, draft-krishnaswamy-dnsop-dnssec-split-view, March 2007.
- [SATIN2011] Hardaker, et al. "Enabling DNSSEC in Open Source Applications." In the proceedings of the SATIN conference (2011).
- [DNSEASY] Mundy, Russ, et al. "DNSSEC Automation and Monitoring." In the proceedings of the DNS EASY Conference, 2011
- [CSRIC] Communications Security, Reliability and Interoperability Council (CSRIC), Working Group 5, DNSSEC Implementation Practices for ISPs (Final Report) (Mar. 22, 2012)
- [RFC6781] O. Kolkman et al. DNSSEC Operational Practices, Version 2. RFC 6781, December 2012

8 List of Symbols, Abbreviations and Acronyms

API	Application Programming Interface
BIND	Berkeley Internet Name Domain Software
CA	Certificate Authority
ccTLD	Country Code Top-Level Domain
CSRIC	FCC Communications Security, Reliability and Interoperability Council
CT	Certificate Transparency
DANE	DNS Based Authentication of Named Entities
DHS S&T	Department of Homeland Security, Science & Technology
DNS	Domain Name System
DNSSEC	Domain Name System Security
EDNS0	Extension Mechanism for DNS
FIPS	Federal Information Processing Standards
gTLD	generic Top Level Domain
ICANN	Internet Corporation for Assigned Names and Numbers
NANOG	North American Network Operators Group
NetSNMP	A simple network management protocol (SNMP) implementation
NTP	Network Time Protocol
OS	Opportunistic Security
RDI	Resources for the DNSSEC Initiative System (the C&A system)
RFC	Request For Comments
SIP	Session Initiation Protocol
SMTP	Simple Mail Transport Protocol
TLD	Top Level Domain
TLS	Transport Layer Security
TLSA	Record used in the DANE protocol in conjunction with TLS